Задача GK? ОПАШКА

Остават броени минути до откриването на новия японски ресторант *Benny’s* в София. Още от ранни зори пред него се е образувала опашка от **N** души, номерирани с числата от **1** до **N**. Един от тях е Алекс, който разбира се не би пропуснал такова събитие. Той предварително е проучил всичко за ресторанта и знае, че не пускат клиентите по реда, в който са се наредили.

Политиката им е доста по-справедлива и се съобразява също с това колко време ще чака всеки един от клиентите. За даден клиент се вземат две неща в предвид – времето, което е чакал на опашката преди да отвори ресторантът – и времето за изпълнение на поръчката му - . Времето, което даден клиент е чакал на опашката, се равнява на сумата от и времето отнело за поръчките на всички клиенти преди него - .

*Benny’s* искат да обслужат клиентите в такъв ред, че максималното време, което е чакал даден клиент, да е минималното възможно. Оказва се обаче, че отношението маси/клиенти е твърде малко и затова ще обслужат само **K** от **N**-те души, като отново се стараят да ги подберат и наредят така, че да е минимизирано времето на чакане.

Алекс повече от всичко би искал да стане част от техния екип и смята да се докаже, като им помогне в избора. За жалост не разполага с много материал за смятане, затова моли Вас да му помогнете, като напишете програма **queue**, която да пресмята търсеното минимално време, което да чака клиентът, стоял на опашката най-дълго.

Вход

На първия ред на стандартния вход се въвеждат естествените числа **N** и **K** – броят души на опашката и колко от тях може да обслужи ресторантът. На следващите **N** реда се въвеждат **N** двойки числа , – времето, което е чакал поредния клиент до отварянето на ресторанта и времето за изпълнение на поръчката му.

Изход

На единствения ред на стандартния изход изведете едно число – търсеното минимално време на чакане.

Ограничения

* 1 ≤ **K** ≤ **N** ≤ 80
* 1 ≤ ≤ 250
* 1 ≤ ≤ 107

Подзадачи

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Подзадача** | **Точки** | **N** | **K** |  |  | **Допълнителни** |
| **1** | 10 | = 2 | = 2 | ≤ 250 | ≤ 107 | - |
| **2** | 5 | = 3 | = 3 | ≤ 250 | ≤ 107 | - |
| **3** | 15 | ≤ 80 | = **N** | ≤ 250 | ≤ 107 | - |
| **4** | 5 | ≤ 80 | ≤ 80 | ≤ 250 | = 1 | - |
| **5** | 35 | ≤ 80 | ≤ 80 | ≤ 250 | ≤ 107 | Съществува оптимално облужване на клиентите, при което всеки следващ е с по-голям номер от предишния. |
| **6** | 5 | ≤ 80 | ≤ 80 | ≤ 250 | ≤ 107 | - |

*Точките за подзадача се получават само ако се преминат успешно всички тестове предвидени за нея.*

**Пример**

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснение** |
| 3 3  1 1  2 3  6 5 | 8 | Оптималният ред би бил: 6 5, 1 1, 2 3. Така първият клиент чака време 6, вторият – 1+5=6, третият – 2+5+1=8. Максималното време, което някой от клиентите е чакал, е 8. |
| **Вход** | **Изход** | **Обяснение** |
| 7 7  7 5  8 5  8 6  6 3  10 8  1 1  5 2 | 31 | Примерен оптимален ред е: 5 2, 6 3, 8 5, 7 5, 8 6, 10 8, 1 1. Тук предпоследният клиент е чакал най-дълго. |
| **Вход** | **Изход** | **Обяснение** |
| 8 4  7 10  9 7  10 5  3 6  2 10  9 3  8 8  3 3 | 13 | Четирите клиента, които ще бъдат обслужени са: 9 3, 10 5, 3 3, 2 10 в този ред. |